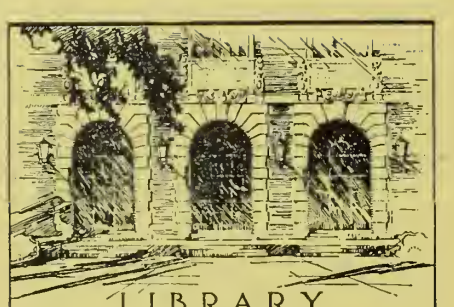

Report No. 201

COO-1018-1077

NEW METHOD OF SOLUTION OF THE TRAVELLING SALESMAN PROBLEM

by

Edward S. Davidson

March 23, 1966

DEPARTMENT OF COMPUTER SCIENCE · UNIVERSITY OF ILLINOIS · URBANA, ILLINOIS

Report No. 201

NEW METHOD OF SOLUTION OF THE TRAVELLING SALESMAN PROBLEM

by

Edward S. Davidson

March 23, 1966

Department of Computer Science
University of Illinois
Urbana, Illinois 61803

Supported in part jointly by the Atomic Energy Commission and the Advanced Research Projects Agency (ARPA) under AEC Contract AT(11-1)-1018.

1. STATEMENT OF THE PROBLEMS

1.1 The Travelling Salesman Problem

A salesman leaves his home in the morning and returns in the evening. He must leave home, visit (n - 1) other cities during the day and return home. Armed with a Rand McNally-type table showing the distance between any pair of these cities, he must calculate the shortest such trip. A trip of this sort is called an n-city tour.

1.2 The Shortest Path Problem

This problem is analogous to the travelling salesman problem except that the necessity for returning home at the end of the day or in fact from starting at home in the morning is removed. Stated directly the salesman is presented with n cities. He may start at any city, and may end at any other city, but must visit each of the remaining (n - 2) cities en route. The total distance travelled, the (n - 1) trips from city to city, is to be a minimum.*

1.3 The Restricted Shortest Path Problem

This problem is analogous to the Shortest Path Problem except that the starting city and the final city are fixed. As defined here an n city Restricted Shortest Path Problem involves (n + 2) cities. One designated 0 is the starting city. One designated (n + 1) is the final city. The remaining n cities are ordered in a path between city 0 and city (n + 1) for minimum total path length.

---

*The shortest path problem is a statement of the backboard wiring problem, in which a set of backboard points are to be electrically common under the restriction that no point may be connected to more than two wires. The goal is to minimize total wire.

2. EQUIVALENCE OF THE THREE PROBLEMS

2.1 Equivalence of Shortest Path Problem and Restricted Shortest Path Problem

Given a Shortest Path Problem, two extra cities may be added: city 0 and city $(n + 1)$. The distance from city 0 or city $(n + 1)$ to any other city is 0. This new problem may be solved as an n city Restricted Shortest Path Problem. The deletion of the extra cities 0 and $(n + 1)$ from the solution will leave the solution to the Shortest Path Problem. Thus an n city Shortest Path Problem is equivalent to an n city Restricted Shortest Path Problem.

Given a Restricted Shortest Path Problem, one can define the distance from city 0 to city $(n + 1)$ arbitrarily since they are never connected. Now add two new cities called -1 and $(n + 2)$. Let $d(x, y)$ be the distance between city x and city y. Assume all distances are non-negative to begin with (otherwise a constant may be added to all distances to make them non-negative).

$$\text{Let } d(0, (n+1)) = \operatorname{Max}_{\substack{x = 0, (n+1) \\ y = 1, \ldots, n}} d(x, y) + 1$$

$$\text{Let } d(-1, 0) = d((n+2), (n+1)) = \operatorname{Min}_{\substack{x, y = 0, \ldots, (n+1) \\ x \neq y}} d(x, y) - 1$$

$$\text{Let } d(-1, i) = d((n+2), j) = 2 \operatorname{Max}_{\substack{x, y = 0, \ldots, (n+1) \\ x \neq y}} d(x, y) + 1 \qquad i = 1, \ldots, (n+2) \quad j = -1, \ldots, n$$

This new problem may be solved as a Shortest Path Problem. The end cities of the minimum solution must be -1 and $(n + 2)$. Also the city next to -1 must be 0 and the city next to $(n + 2)$ must be $(n + 1)$. Otherwise a change in path could be found to decrease the length of the solution. Thus, deleting cities -1 and $(n + 2)$ from the solution, the solution to the Restricted Shortest Path Problem has been found. Thus an n city Restricted Shortest Path Problem is equivalent to an $(n + 4)$ city Shortest Path Problem. Hence the Restricted Shortest Path Problem is equivalent to the Shortest Path Problem.

Q.E.D.

2.2 Equivalence of the Travelling Salesman Problem and the Restricted Shortest Path Problem

Given a Travelling Salesman Problem one can pick an arbitrary city, designated "city 0", and duplicate it designating its duplicate "city n". This new problem can be solved as a Restricted Shortest Path Problem. After the solution has been found the cities 0 and n can be reunited forming the shortest tour (solution to the Travelling Salesman Problem). Thus an n city Travelling Salesman Problem is equivalent to an (n - 1) city Restricted Shortest Path Problem.

Given a Restricted Shortest Path Problem,

let $d'(0, n+1) = \underset{x=1,\ldots n}{\text{Min}}\, d(x, 0) - 1 + \underset{x=1,\ldots n}{\text{Min}}\, d(x, (n+1)) - \underset{x,y=1,\ldots n}{\text{Max}}\, d'(x, y)$.

This insures that this problem can be solved as a Travelling Salesman Problem and that cities 0 and (n + 1) will be connected in the minimum solution. Otherwise a change in tour could be found which would decrease the length of the tour. Separating cities 0 and (n + 1) to form a path provides the solution to the Restricted Shortest Path Problem. Thus an n city Restricted Shortest Path Problem is equivalent to an (n + 2) city Travelling Salesman Problem. Hence the Travelling Salesman Problem and the Restricted Shortest Path Problem are equivalent. Q.E.D.

So, finally, all three of the problems are equivalent. Hence any techniques applicable to any one of these problems is useful in the solution of any other. It is the purpose of this paper to present a solution to the Restricted Shortest Path Problem.

3. PRELIMINARY DISCUSSION

For the remainder of this paper the mere classical terminology is dropped: cities are referred to as points and paths between pairs of cities as wires.

3.1 Solutions

A solution is a finite set of distinct wires [each of which may be named by the unordered pair of its end points, e.g. (a, b)] possessing the following two properties:

i) points 0 and (n + 1) appear exactly once each as end points in the set of wires,

ii) all other points, i.e. 1, 2, ...n, appear exactly twice each as end points in the set of wires.

A solution then forms a path from 0 to (n + 1) with some, or possibly all, of the other points between in some order. The remainder of the points form a set of cycles, where each cycle has no points in common with the other cycles or with the path: see Fig. 1. Since all the wires in the set are distinct, each cycle must contain at least three points.

Figure 1: Example of a 9-City Solution with Two Cycles

Solutions containing no cycles are desired and are called feasible solutions. Solutions which contain cycles are called infeasible solutions.

The cost of a solution is the sum of the lengths of the wires in the solution. The minimum cost feasible solution is to be found.*

3.2 Subtours

A "subtour", or a "disjoint set of subtours", operates on a given

*The minimization algorithm to be developed does not imply that "length" of a wire is geometrically determined. These lengths can be arbitrary real numbers.

solution to form another solution.

A subtour is defined as an ordered list, $(p_1, p_2, \cdots p_{2m})$, of some of the points: 0, 1, 2, ..., (n + 1). The list must contain an even number of entries. The wires (p_1, p_2), (p_3, p_4), ..., (p_{2m-1}, p_{2m}) are to be deleted from the given solution; the wires (p_2, p_3), (p_4, p_5), ... (P_{2m-2}, p_{2m-1}), (p_{2m}, p_1) are to be added to the given solution. Each of these deleted and added wires must be unique. Clearly wires to be deleted must have appeared in the given solution and wires to be added must not have appeared in the given solution. The new solution is found by performing deletions and additions on the given solution as indicated by the subtour. The order in which these deletions and additions are performed is immaterial.

A given point may appear in the list once, twice, or not at all. No point may appear more than twice in a subtour since no more than two wires with that point as an end point may be deleted from the given solution. Similarly points 0 and (n + 1) may not appear more than once each.

A subtour may be represented as in Fig. 2. A (-) represents a wire to be deleted and a (+) represents a wire to be added. Remember that although the p_i's are not necessarily distinct, the wires are distinct.

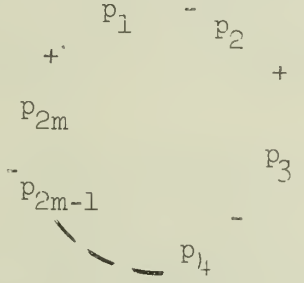

Figure 2:- Representation of Subtour $(p_1, p_2, p_3, p_4, \cdots p_{2m-1}, p_{2m})$

A disjoint set of subtours is a set of subtours such that no two subtours have any points in common. A necessary consequence is that no two subtours have any wires in common. A given solution may be mapped onto a new solution by a disjoint set of subtours. The new solution is found by performing deletions and additions on the given solution as indicated by each of the subtours in the set, successively. Again the order in which the deletions and additions are performed as well as the order in which the subtours are performed is immaterial.

It must be realized that a subtour, or disjoint set of subtours, does not necessarily map a feasible solution onto another feasible solution, but rather maps a solution onto another solution: see Fig. 3. The problem of infeasible solutions appears endemic to several other methods of solution.

0 - 1 - 2 - 3 - 4 - 5 - 6 - 7

Figure 3a. Original Solution

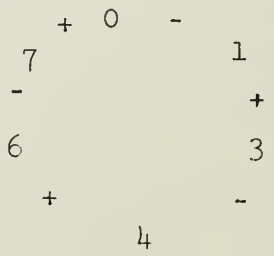

Figure 3b. Subtour

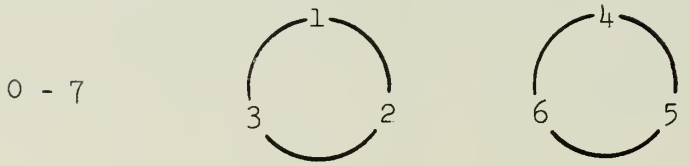

Figure 3c. Final Solution

The cost of a subtour is the sum of the added (+) wire lengths minus the sum of the deleted (-) wire lengths. The cost of a disjoint set of subtours is the sum of the costs of the subtours of the set. Thus the change in cost of a solution realized by performing a subtour, or disjoint set of subtours, is the cost of that subtour or set of subtours. A negative subtour is a subtour with negative cost, i.e. a subtour which diminishes the cost of a given solution. A feasible disjoint set of subtours is a disjoint set of subtours which operates on a feasible solution to form another feasible solution.

4. THEOREM 1: PERMUTATION THEOREM

The mapping from a given feasible solution onto another feasible solution may be realized by operating on the given solution with a disjoint set of subtours.

Proof: Such a mapping may be represented by a permutation of the points 1 through n in the path from 0 to (n + 1). This permutation determines a set, D, of deleted wires and a set, A, of added wires. Each end point which appears 0, 1, or 2 times in D appears 0, 1, or 2 times, respectively, in A. Pick any wire from D, e.g. (p_1, p_2). Find a wire in A with p_2 as an end point, e.g. (p_2, p_3). Find a wire in D with p_3 as an end point, e.g. (p_3, p_4). Continue choosing wires alternately in this fashion until some wire is chosen from A or D with p_1 as an end point, e.g. (p_m, p_1). The existence of such a wire in A is guaranteed since (p_1, p_2) was a wire in D. Observe that if some other point $p_i \neq p_1$ appears as an end point of its third wire, p_i must appear twice in both D and A. Thus a fourth wire can be found in D or A with p_i as an end point allowing the process of selecting wires to continue until (p_m, p_1) is chosen. If a wire of the form (p_m, p_1) is first chosen from D, rather than A, there must yet be two appearances of p_1 as an end point in A, one of which will be chosen next and the other of which will be chosen later to complete the subtour.

Thus the process can be continued until a subtour $(p_1, p_2, \cdots, p_m)$ is formed. All points still appear an equal number of times in the remaining wires of D and A. Thus the process can be repeated on the remaining elements of D and A until no elements are left in A (or D). A set of subtours has now been formed which represents the permutation, but which may not be disjoint.

Suppose some point p_i appears in two subtours: $(p_1, p_2 \cdots p_{i-1}, p_i\ p_{i+1}, \cdots p_m)$ and $(q_1, q_2 \cdots q_j, p_i, q_{j+1} \cdots q_\ell)$. Suppose (p_i, p_{i+1}) and (p_i, q_j) were elements of D. This assumption can be made with full generality. Construct a new subtour:

$$(p_i, p_{i+1}, \cdots, p_m, p_1, p_2, \cdots, p_{i-1}, p_i, q_j, \cdots q_2, q_1, q_\ell \cdots q_{j+1})$$

which replaces the two previous subtours: see Fig. 4. This process of combination of non-disjoint subtours can be repeated as often as necessary until all subtours in the set are disjoint. Q.E.D.

From this theorem if follows that a given solution will be mapped onto the optimum solution by operating on the given solution with the most negative feasible disjoint set of subtours.

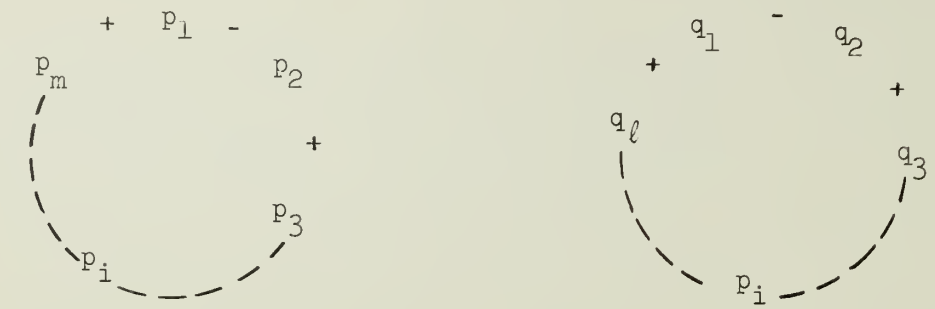

Figure 4a. Two Non-Disjoint Subtours

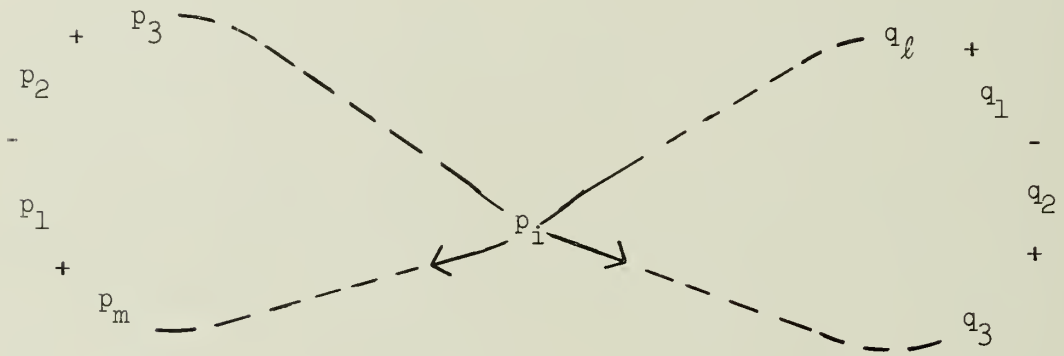

Figure 4b. New Combined Subtour

5. THEOREM 2: CIRCLE THEOREM

Consider a finite sequence of negative and non-negative real numbers the sum of which is negative. Let these numbers be arrayed on the circumference of a circle, so that the successor of the last number is the first. For each number in the circle a sequence of partial sums can be built up consisting of the number itself, the sum of the number and its clockwise successor, the sum of the previous partial sum and the next clockwise successor, ...until that number is added whose successor is the first number. The theorem statement is that in the circle there is at least one number all of whose partial sums are negative.

Proof: There is at least one negative number in a set of numbers whose sum is negative. If there are no non-negative numbers any number may be a starting number. Otherwise find all negative numbers whose clockwise successor is non-negative. Add these numbers to their clockwise successors and replace the two numbers by their sum in the circle. The sum of the numbers in the new circle is the same as the sum of the numbers in the old circle; but there are fewer numbers in the new circle.

Apply this process repetitively until all the numbers in the circle are negative or the circle has been replaced by one negative number. One of these situations must occur since each new circle contains fewer elements than the previous circle and contains at least one negative number, and since the process can continue as long as there are non-negative numbers in the circle.

For each negative number in the final circle there is a possible starting number for the theorem. Each negative number in the final circle may be said to represent a unique segment (ordered list of numbers) of the original circle. The set of these segments represents all the numbers of of the original circle. Starting with the leftmost number of any segment and proceeding clockwise around the circle, all partial sums will be found to be negative. Q.E.D.

It follows that all negative subtours can be found (a restricted case of the theorem) by building up all sequences of alternating deleted

and added wires, after the fashion of a subtour. Sequences are built up one wire at a time such that the sum of the added wire lengths minus the sum of the deleted wire lengths remains negative throughout. Sequences which are positive need never be considered.

6. PROPOSED METHOD OF SOLUTION OF THE TRAVELLING SALESMAN PROBLEM

It is expected that the majority of time consumed in finding solutions by this method will be spent finding negative subtours. Therefore it is desirable to find all the negative subtours before actually performing any subtours. This procedure will allow convergence to optimum in one iteration. Therefore some effort should be made to find a starting solution fairly close to optimum.

A suitable starting solution is found in Phase I. Find a first solution and iteratively apply a suboptimal convergence technique proposed by Croes[2]. Croes' technique amounts to performing feasible subtours of length four: i.e. removing and adding two wires at a time. These subtours are quite easy to find and successfully remove obvious non-optimalities from the solution. The final solution of Phase I is the starting solution.

In Phase II one considers the starting solution and finds all negative subtours by application of the circle theorem.

In Phase III one finds all disjoint sets of one or more of the subtours (from Phase II) and orders these sets by negativity of cost, with the most negative first. Starting at the top of the list each subtour set is labelled "feasible" or "infeasible" according to whether the solution it produces is feasible or infeasible. The labelling is carried out until the first feasible subtour set is found, or the set is exhausted. In the latter case the empty subtour set is appended to the bottom of the list and is given cost zero. Unlabelled subtour sets are discarded. Remaining then are a number of infeasible subtour sets and one feasible subtour set.

The purpose of Phase IV is to determine if any of the infeasible subtour sets can be made feasible by the addition of positive subtours. These positive subtours must be disjoint from the subtour sets to which they are to be added and must not increase the cost of the combined set to more than the cost of the feasible set already at hand. Positive subtours are found which are disjoint from one or more of the infeasible subtour sets. Exactly those are found which, if added to one of the infeasible subtour sets, would not make the cost of the combined set greater than the cost of the

feasible subtour set.

In Phase V each infeasible subtour set is considered in turn and combined in as many disjoint ways as possible with the positive subtours (found in Phase IV), keeping cost of the combined set less than that of the feasible subtour set. If any of these are found to be infeasible they are discarded. If one of these is found to be feasible, it becomes the new feasible subtour set and the old feasible subtour set is discarded. The process continues until only one feasible subtour set remains. By application of the permutation theorem, the original solution operated on by this feasible subtour set produces the optimum solution.

6.1 Phase I

Probably the most easily found good first solution involves picking an arbitrary first point. The next point is the closest point to the first point. Now the partial solution has a left and a right end point. Of the remaining points consider the one closest to the left end point and the one closest to the right end point. The closer of these two becomes the new left or right end point respectively. Continue until all points are in the path. This is the first solution. Points 0 and $(n + 1)$ are now added to the left and right end points respectively. This is a common technique which may be found presented in full generality, for example, in the article by Loberman and Weinberger[8].

Perform Croes'[2] "desirable inversions" on the first solution until no more are possible. An inversion (ℓ, r) is the subtour $[(\ell - 1), \ell, (r + 1), r]$ which maps the initial solution:

$$0 1 \ldots (\ell - 1)\ \ell\ (\ell + 1) \ldots (r - 1)\ r (r + 1) \ldots n(n + 1)$$ onto the solution:

$$0 1 \ldots (\ell - 1)\ r (r - 1) \ldots (\ell + 1)\ \ell (r + 1) \ldots n(n + 1).$$

An inversion (ℓ, r) is the same as the inversion $[(r + 1), (\ell - 1)]$ which has subtour $[r, (r + 1), \ell, (\ell - 1)]$ which is easily seen to be equivalent to the subtour for (ℓ, r). Thus, the set of all inversions is equivalent to the set of all inversions (ℓ, r) such that $r > \ell$. Desirable inversions are those which have negative subtours. To implement this procedure, for example, the $n(n - 1)/2$ possible inversions are examined in turn until the

first desirable inversion is found. This inversion is performed and scanning for desirable inversions resumes again. This process is continued until one complete scan produces no desirable inversions. Modifications of this technique should be considered to produce a starting solution which is sufficiently close to optimum without large expense of time.

6.2 Phase II

Find all negative subtours by utilizing the circle theorem. Subtour segments are built up from basic segments a, b, c, [i.e. delete the wire (a, b) and add the wire (b, c)]. The next segment would be of the form c, d, e which would form a subtour if e = a. Basic segments are added only if the wires are different from those already chosen and only if the partial sum stays negative.

All basic segments are listed for each point as first point, e.g. for point 5:

(5, 6, 0), ..., (5, 6, 4), (5, 6, 8), ...,(5, 6, (n + 1)),

(5, 4, 0), ...,(5, 4, 2), (5, 4, 6), ..., (5, 4, (n + 1)).

These segments are listed for each point in order of desirability, most negative first.

Phase II can be programmed by a backtrack programming algorithm[4] for tree running. Each subtour segment must eventually produce zero or one subtours and reach a point where a second subtour is completed or no possible segments may be added due to the necessity of deleting or adding the same wire twice or making the partial sum non-negative. When a subtour segment reaches this situation, it is dropped from further consideration. By far the majority of partial subtours are discarded in the general case before much calculation is done.

6.3 Phase III

The set of negative subtours is augmented by all sets of two or more of these subtours which are disjoint. The set with the most negative cost is then examined for feasibility. If it is feasible, it must lead to optimum due to the permutation theorem. If it is not feasible, the list of

negative subtour sets is scanned until the first feasible subtour set is found or the list is exhausted. The first feasible subtour set is listed last and the remaining subtour sets, if any, are discarded. If not feasible subtour set is in the list, the empty subtour set is appended to the bottom of the list and is assigned cost zero. The empty subtour set is by definition feasible.

6.4 Phase IV

Phase IV is quite similar in nature to Phase II. Let the list of subtour sets from Phase III be represented by IF_1, IF_2, ...IF_K, F. Also let the cost of IF_j be $C(IF_j)$. Subtours are now sought which are disjoint from IF_1 keeping partial costs less than $C(F) - C(IF_1)$ instead of keeping them negative as in Phase II.

If a partially constructed subtour becomes non-disjoint from IF_1 the smallest j is sought such that it is disjoint from IF_j. The "cost bound" for the subtour is appropriately reduced from $C(F) - C(IF_1)$ to $C(F) - C(IF_j)$. Each positive subtour found is tagged with the j which was in use when the subtour was completed. If a partially constructed subtour becomes non-disjoint from IF_K, it is discarded.

6.5 Phase V

Consider IF_1 in combination with every disjoint set of the positive subtours tagged with $j = 1$. The cost of the combined set must be less than $C(F)$ otherwise it is discarded. Examine the remaining sets for feasibility. If any are found to be feasible, the one with least cost replaces F and its cost becomes $C(F)$. Examine each of the other IF's in turn. Observe that IF_j must be combined with every disjoint set of positive subtours, tagged j or less, from which it is disjoint. The remainder of the analysis is the same as the above. After IF_K has been considered, the set designated F will convert the given solution to the optimum solution by the permutation theorem. The most negative disjoint feasible set of subtours has been found.

7. CONCLUSION

Many methods have been proposed for the solution of the Travelling Salesman Problem. Of the methods for which the amount of computation is invariant to the actual data (other than the number of points) involved, the dynamic programming approach[1, 5, 7] is the most practical. The amount of computation and storage requirements, however, becomes quite large for problems of more than about thirteen points, if an optimum solution is desired.

Of the "data sensitive" methods, Little's[6] method seems to have met with more success than the dynamic programming methods in certain problems of a large number of points. Little's method also works quite well for smaller problems. Croes'[2] "inversion" technique is extremely simple and requires few computations, but it is non-optimal.

This paper has presented a generalization of Croes' technique which leads to optimum solutions. Since the method presented here has not yet been programmed, little can be said about its comparative efficiency. In cases where this method takes excessive amounts of time, however, the Phase I or Phase III feasible solution can be accepted, though it is possibly non-optimal.

8. ACKNOWLEDGEMENT

Several people have provided guidance in the formulation of this plan of attack on the Travelling Salesman Problem. In particular, John A. Wilber, now at Automatic Electric Laboratories, invested considerable amounts of his time and energy at each stage of development. I also wish to express thanks to Professor Bruce H. McCormick whose advice was greatly appreciated.

9. REFERENCES

1. Bellman, R., "Dynamic Programming Treatment of the Travelling Salesman Problem", J. Assoc. Comput. Mach., 9 (1962), pp. 61-63.

2. Croes, G. A., "A Method for Solving Travelling-Salesman Problems", Operations Res., 6 (1958). pp. 791-812.

3. Dantzig, G. B., Linear Programming and Extensions, Princeton Univ. Press, Princeton, N. J., (1963), pp. 545-547.

4. Golomb, S. W. and Baumert, L. D., "Backtrack Programming", J. Assoc. Comput. Mach., 4 (1965), pp. 516-524.

5. Held, M. and Karp, R. M., "A Dynamic Programming Approach to Sequencing Problems", J. Soc. Indust. Appl. Math., 1 (1962), pp. 196-210.

6. Little, J. D. C., et al., "An Algorithm for the Travelling Salesman Problem", Operations Res., (1963), pp. 972-989.

7. Lin, S., "Computer Solutions of the Travelling Salesman Problem", Bell Syst. Tech. Jour., (1965), pp. 2245-2269.

8. Loberman, H. and Weinberger, A., "Formal Procedures for Connecting Terminals with a Minimum Total Wire Length", J. Assoc. Comput. Mach., (1957), pp. 428-437.